A Complete Axiomatization of Knowledge and Cryptography

Mika Cohen Mads Dam
KTH Computer Science and Communication
Stockholm, Sweden
{mikac,mfd}@nada.kth.se

Abstract

The combination of first-order epistemic logic with formal cryptography offers a potentially powerful framework for security protocol verification. In this paper, cryptography is modelled using private constants and one-way computable operations, as in the Applied Pi-calculus. To give the concept of knowledge a computational justification, we propose a generalized Kripke semantics that uses permutations on the underlying domain of cryptographic messages to reflect agents' limited resources. This interpretation links the logic tightly to static equivalence, another important concept of knowledge that has recently been examined in the security protocol literature, and for which there are strong computational soundness results. We exhibit an axiomatization which is sound and complete relative to the underlying theory of terms, and to an omega-rule for quantifiers. Besides standard axioms and rules, the axiomatization includes novel axioms for the interaction between knowledge and cryptography. As protocol examples we use mixes, a Crowds-style protocol, and electronic payments. Furthermore, we provide embedding results for BAN and SVO.

1 Introduction

Many goals of cryptographic communication concern knowledge. Authenticity, for instance, may mean that a receiver knows the sender of a message, and anonymity may mean that the sender is unknown to an eavesdropper. In these contexts, knowledge is used as a semantical rather than a cognitive concept: The intention is that local interactions (for instance, the local observations of the receiver) force some global system property (for instance, that the message has a certain sender).

Knowledge, in this sense, is often tied to the concept of indistinguishability. This applies, for instance, in security protocol analysis using some form of observational equivalence (cf. [17]), in multi-agent system semantics using local history identity (cf. [16, 27]), and in information flow theory using low-level observability (cf. [31]). For cryptographic communication the definition of a suitable indistinguishability relation is somewhat delicate. The baseline is computational indistinguishability in the sense of modern cryptography (cf. [20]), i.e., the absence of a computationally feasible experiment to tell two ciphertexts apart. On the other hand, in order to serve as a basis for a useful logic, the relation should be amenable to formal treatment.

To this end, static equivalence [17] has recently emerged as a natural starting point. Static equivalence collects cryptographic terms that are visible to an agent (or the environment) in a frame, roughly a sequence s of terms. Two frames s and s' are equivalent if they satisfy the same equality tests. For instance, if *encrypt*$(s(i), s(j)) = s(k)$ then *encrypt*$(s'(i), s'(j)) = s'(k)$, where $s(i)$ is the i:th term in sequence s. Static equivalence is parametrized on an underlying equational theory of cryptographic terms over a signature of "feasibly computable" operators. Depending on the specific choice of theory, strong links between static equivalence and computational indistinguishability can sometimes be established. Computational soundness, static equivalence implying computational indistinguishability, has recently received particular attention (cf. [1]).

In this paper, we use static equivalence as the basis for building a program logic. Our first step is the observation that static equivalence implicitly defines a correspondence between messages, with messages deduced by the same sequence of computations corresponding to each other: Message *encrypt*$(s(i), s(j))$ at frame s corresponds to message *encrypt*$(s'(i), s'(j))$ at frame s', and so on. This correspondence can be lifted to assignments V of messages to variables $x, y, z, ...$ of a logic: V at s corresponds to V' at s' if $V(x) =$ *encrypt*$(s(i), s(j))$ implies that $V'(x) =$ *encrypt*$(s'(i), s'(j))$, and so on. Motivated by this observation, we instantiate the multi-agent system framework [16], using frames as states and sub-frames as local states.We say that property F is known to agent A at global state s under assignment V, $s, V \models \Box_A F$, if and only if $s', V' \models F$ for all global states s' and assignments V' such that agent A's local state in s is statically equivalent to A's local state in

s', and V at A's local state in s corresponds to V' at A's local state in s'. This idea follows counterpart semantics [24] by checking F at s' under a corresponding assignment V', rather than under the original assignment V. This interpretation has a number of interesting properties.

Firstly, our use of counterpart semantics provides a handle on the difficult issue of mathematical omniscience in epistemic logic. Existing multi-agent system semantics (cf [5, 21, 30, 32]) follow basic Kripke semantics [9], where the assignment V is kept constant as indistinguishable states are scanned: $s, V \models \Box_A F$, if and only if $s', V \models F$ for all indistinguishable states s'. Assuming mathematical equalities depend only on the assignment V, and not on the global state s, agents are *cryptographically omniscient*, i.e., know all equalities:

$$t = t' \rightarrow \Box_A\, t = t' \tag{1}$$

for any open terms t and t' built from cryptographic operators f and variables x. For example, $x = \mathit{decrypt}(y, z) \rightarrow \Box_A x = \mathit{decrypt}(y, z)$ holds even when agent A has not obtained the key z. Thus, the epistemic modality is vacuous on cryptographic statements. Various suggestions, based on explicit knowledge [16], have been made towards addressing this issue (cf. [6, 22, 25, 28]). However, work on explicit knowledge abandon Kripke-style semantics, lose many of the useful logical properties of knowledge, and provide a very syntactic account of knowledge extraction.

By contrast, we avoid cryptographic omniscience (1) even though we stay within a Kripke-style semantics and preserve most logical properties of knowledge. This is shown by the second result, which is also the main result of the paper: A sound and complete axiomatization, based on axioms and rules from standard first-order logic and modal S5 logic. In addition, the axiomatization includes some novel axioms for the interaction between knowledge and cryptography. The first interaction axiom weakens (1):

$$A\ \mathit{deduces}\ \overline{x} \rightarrow (y = f(\overline{x}) \rightarrow \Box_A y = f(\overline{x}))$$

for each feasibly computable operator f. The predicate *deduces* is Dolev-Yao-style message deduction in the sense of [17], and is definable in terms of the epistemic modality, as will be explained below. Another interaction axiom says that the agent knows a property of non-deduced values $\overline{x}$ only if this property holds of any non-deduced values $\overline{z}$:

$$\neg A\ \mathit{deduces}\ \overline{x}, \overline{z} \rightarrow \Box_A F \rightarrow F[\overline{z}/\overline{x}]$$

In order to obtain completeness with respect to any given theory of abstract cryptography, the axiomatization uses some infinitary machinery. Firstly, we add as axioms all equalities and inequalities from the underlying theory of cryptographic terms. Secondly, we add a second kind of quantifiers, $\forall m$-quantifiers, with an omega-rule. Intuitively, the formula $\forall m.F[m/x]$ expresses the infinite conjunction $F[M_1/x] \wedge F[M_2/x] \wedge ...$, where $M_1, M_2, ...$ lists all ground message terms. As explained in section 10, the completeness result significantly improves on our earlier result for a propositional epistemic logic [12].

The third result is epistemic characterizations of message deduction and static equivalence. For the former, we obtain $A\ \mathit{deduces}\ x \leftrightarrow \exists y.\Box_A y = h(x)$, where h is a one-way hash operator (cf. [2] for a related use of the hash operator). The logical characterization of static equivalence, which is rather immediate, gives added credence to our semantics, and allows the transfer of computational soundness results, such as that of [1], to our epistemic logic. It follows, for instance, that if the same properties are known by agent A in global states s and s' then A's local states in s and s' are computationally indistinguishable.

We illustrate the language in three example protocols, using mixes, a Crowds style protocol [29], and electronic payments. Furthermore, we illustrate the axiomatization by embedding characteristic rules from authentication logics BAN [10] and SVO [32], including an infinitary weakening of necessitation appropriate for BAN. The protocol specifications and the embedding results (as well as the characterization of message deduction above) all rely on the absence of cryptographic omniscience.

All proofs appear in [14].

2 Messages and Static Equivalence

Let f range over a countable set Σ of public, feasibly computable operators, each equipped with an arity. Let $A, B, ...$ range over a finite, non-empty set $\mathcal{A} \subseteq \Sigma$ of 0-arity operators, representing public names of distinct agents; Other 0-arity operators in Σ also represent public values, "plain texts". Let c range over a countably infinite set *SEC* of secret constants, and $x, y, z...$ range over a countably infinite set *VAR* of variables. Message terms t are:

$$t ::= x \mid c \mid f(t_1, ..., t_n)$$

where f has arity n. Write $\mathit{VAR}(t)$ for the set of variables in t. Let $M, K, N, ...$ range over the set $\mathcal{T}$ of ground terms (terms with no occurrences of variables). An abstract model of cryptography is given as a congruence $\equiv$ over ground terms, typically via an equational theory. The set of messages is the set $\mathcal{T}_{\equiv}$ of all equivalence classes with respect to $\equiv$. Overloading notation, we write M for the equivalence class $[M]_{\equiv}$, and f for its induced operation on classes.

Example 1 *To model pairing and asymmetric encryption, we assume the least congruence over ground terms satisfying* $\mathit{fst}(\mathit{pair}(M, M')) \equiv M$, $\mathit{snd}(\mathit{pair}(M, M')) \equiv M'$ *and* $\mathit{dec}(\mathit{enc}(M, \mathit{pk}(K)), K) \equiv M$. *Informally, fst/snd picks out first/second components, pk derives a public key from a private key, and* enc/dec *encrypts/decrypts the first argument using the second as key.*

Throughout this paper, we assume that agent names in $\mathcal{A}$ are non-equivalent. In some results, we assume there is a special unary operator $h \in \Sigma$, with $h(h(M)) \not\equiv M$ and such that if $h(M) \equiv h(M')$ then $M \equiv M'$; We call such an operator a *hash operator*.

Assume a non-empty, countable set *LOC* of store locations. A state ("store") over *LOC* is a partial function s from *LOC* to $\mathcal{T}_{\equiv}$. A message is inferable ("deducible") from a state if the message is directly given by the state, i.e., belongs to the range, or if the message can be obtained from already inferred messages through some $f \in \Sigma$.

Definition 1 *Inferable*(s)*, the messages inferable from* s*, is the least extension of ran*(s) *closed under all* $f \in \Sigma$*.*

Constant c need not be in *Inferable*(s), but 0-arity f must.

We introduce a second kind of terms, s-terms:

$$\alpha ::= l \mid f(\alpha_1, ..., \alpha_n)$$

where $l \in dom(s)$ and $f \in \Sigma$. Each s-term represents an inference path available at s. We extend s to a mapping on s-terms, i.e., $s(f(\alpha_1, ..., \alpha_n) = f(s(\alpha_1), ..., s(\alpha_n))$. The following corollary corresponds to proposition 1 in [2].

Corollary 1 *Inferable*$(s) = \{s(\alpha) : \alpha \in$ *s-terms*$\}$.

Two states are statically equivalent if they satisfy the same equality tests:

Definition 2 *States* s *and* s' *are statically equivalent, written* $s \approx s'$*, if and only if,* $dom(s) = dom(s')$ *and:*

$$s(\alpha) = s(\alpha') \Leftrightarrow s'(\alpha) = s'(\alpha'), \textit{ all } s\textit{-terms } \alpha, \alpha'$$

In relation to [17], constants c correspond to private/fresh names, states s correspond to frames, *Inferable*(s) corresponds to message deduction from the frame s, and $s \approx s'$ is static equivalence between (finite) frames s and s'.

3 Indistinguishability under Permutation

To fit in a counterpart semantics, we reformulate static equivalence in a manner strongly reminiscent of framed bisimulation [3]. Assuming $s \approx s'$, the message $s(\alpha)$ at s corresponds to the message $s'(\alpha)$ at s' in that both messages are reached through the same inference path. Motivated by this intuition, we introduce an indistinguishability $\sim$ between states, which is relativized to a permutation ρ on $\mathcal{T}_{\equiv}$. Informally, if $s \sim^{\rho} s'$, then any message M at s corresponds to $\rho(M)$ at s'. To qualify as a witness for state indistinguishability, a permutation ρ must respect locations as well as all operations in Σ on inferable messages:

Definition 3 $s \sim^{\rho} s'$*, if and only if,* $dom(s) = dom(s')$ *and:*

- $\rho \circ s = s'$.
- $\rho(f(\overline{M})) = f(\overline{\rho(M)})$*, if all* $M_i \in$ *Inferable*(s).

Lemma 1 *If* $s \sim^{\rho} s'$ *then* $\rho($*Inferable*$(s)) =$ *Inferable*(s').

Proposition 1 *The following hold:*

1. $s \sim^{Id} s$
2. *If* $s \sim^{\rho} s'$ *and* $s' \sim^{\rho'} s''$ *then* $s \sim^{\rho' \circ \rho} s''$.
3. *If* $s \sim^{\rho} s'$ *then* $s' \sim^{\rho^{-1}} s$.

Write $\overline{\textit{Inferable}(s)}$ for the complement of *Inferable*(s). Messages in $\overline{\textit{Inferable}(s)}$ are anonymous in that every permutation of $\overline{\textit{Inferable}(s)}$ is "epistemically possible":

Corollary 2 *Assume a permutation* π *on* $\overline{\textit{Inferable}(s)}$*. Extend* π *to a permutation* ρ *on* $\mathcal{T}_{\equiv}$ *such that* $\rho(M) = M$ *for* $M \in$ *Inferable*(s)*. Then,* $s \sim^{\rho} s$.

A state s is *normal* if s has countably infinite many non-inferred messages, i.e., $\overline{\textit{Inferable}(s)}$ is countably infinite. This corresponds to the assumption in [17] that there always are fresh private names available. In the following two results, relating $\sim$ to $\approx$, we assume states are normal.

Lemma 2 $s \sim^{\rho} s'$ *if, and only if,* $dom(s) = dom(s')$ *and* $\rho(s(\alpha)) = s'(\alpha)$ *for all* s*-terms* α.

Theorem 1 $s \approx s'$*, if and only if,* $\exists \rho : s \sim^{\rho} s'$.

4 Systems and Statements

Multi-Agent System We instantiate the multi-agent system framework [16] to our notion of state. A state space is a non-empty set S of states s over *LOC*, intuitively the set of possible states of some underlying program. An agent projection $|$ assigns a set $\textit{LOC}|A \subseteq \textit{LOC}$ of locations observed (accessed) by agent A. The agent projection is lifted to states: $s|A$ is the restriction of s to locations in $\textit{LOC}|A$. A multi-agent system, or simply a system, is a structure $\mathcal{S} = \langle \textit{LOC}, S, | \rangle$ of a set *LOC* of store locations, a state space S and an agent projection $|$.

Example 2 *We model a system where either agent* A *or agent* B *posts a message, but agent* C *cannot observe whom. Assume the message congruence from example 1. Assume two locations:* $\textit{LOC} = \{\textit{sender}, \textit{post}\}$*. The state space is* $S = \{s : \textit{LOC} \to \mathcal{T}_{\equiv} \mid s(\textit{sender}) \in \{A, B\}\}$*. Agent* C *observes only the post location:* $\textit{LOC}|C = \{\textit{post}\}$*. The system is* $\mathcal{S} = \langle \textit{LOC}, S, | \rangle$.

Inference and indistinguishability naturally relativize to an agent A: *Inferable*$(A, s) =_{df}$ *Inferable*$(s|A)$; $s \sim_A^{\rho} s'$, if and only if, $s|A \sim^{\rho} s'|A$.

Statements Statements $F \in \mathcal{F}$ are defined by:

$$F \quad ::= \quad t = t' \mid p(t_1, ..., t_n) \mid \forall x.F \mid \forall m.F[m/x] \mid \Box_A F \mid F \wedge F' \mid \neg F$$

where $A \in \mathcal{A}$, $x \in \mathit{VAR}$, p is from a countable set $\mathcal{P}$ of predicates, m is from a countably infinite set of "place holders", and $F[m/x]$ is the result of uniformly replacing free occurrences of variable x by place holder m throughout F. Note that a statement may contain unbound variables, but not unbound place holders.

The language has two types of quantifier, reflecting the *de re*/*de dicto* dichotomy familiar from first-order modal logic [9]. To explain the distinction, say that A receives the value $\mathit{enc}(c, c')$, where either of c, c' may be unknown to A. Is it then true that "A knows that A received $\mathit{enc}(c, c')$"? Under one interpretation, the *de re* interpretation, the answer is yes: The value ("bitstring") denoted by $enc(c, c')$ is known by A to be received. Under the *de dicto* interpretation, on the other hand, the statement is about the term "$\mathit{enc}(c, c')$" itself. In this case, the statement might be false: Agent A need not know that the term used, "$enc(c, c')$", applies to the value received. In our language, variables $x \in \mathit{VAR}$ refer *de re*, while closed terms $M \in \mathcal{T}$ refer *de dicto*. To illustrate, the *de re* statement $\forall x.(A \mathit{\ received\ } x \rightarrow \Box_A A \mathit{\ received\ } x)$ is intuitively valid, while the corresponding *de dicto* statement $\bigwedge_{M \in \mathcal{T}} (A \mathit{\ received\ } M \rightarrow \Box_A A \mathit{\ received\ } M)$ is intuitively invalid. Of course, the latter statement is not part of the language, since there are no infinite conjunctions. However, in our language, the $\forall m$-quantifier is used for such quantification over closed terms: The statement $\forall m.F[m/x]$ expresses the conjunction $\bigwedge_{M \in \mathcal{T}} F[M/x]$. To highlight their respective use, we will refer to the $\forall x$-quantifier and the $\forall m$-quantifier as, respectively, the *de re* quantifier and the *de dicto* quantifier. Although we believe that the use of the dicto quantifier is of independent interest, its motivation here is mainly technical. To obtain a complete axiomatization, we need an axiom stating that each variable x refers to some message M. Using the *de dicto* quantifier, we can express this grounding by the statement $\exists m.x = m$. In section 11, we show that the *de dicto* quantifier cannot be reduced to the *de re* quantifier.

Interpreted System A predicate interpretation I on a system $\mathcal{S}$ assigns, to each predicate p and state $s \in S$, a relation $I(p, s)$ in $\mathcal{T}_{\equiv}$ (matching the arity of p). An interpreted system based on a system $\mathcal{S} = \langle \mathit{LOC}, S, | \rangle$ is a structure $\mathcal{I} = \langle \mathit{LOC}, S, |, I \rangle$ where I is an interpretation on $\mathcal{S}$. In some examples and propositions, we explicitly introduce the special unary predicates $A \mathit{\ infers}$ and $@_l$, for $A \in \mathcal{A}$ and $l \in \mathit{LOC}$. When we do so, we implicitly require that $I(A \mathit{\ infers}, s) = \mathit{Inferable}(A, s)$ and $I(@_l, s) = \{s(l)\}$.

5 Cryptographic Counterpart Semantics

In this section, we interpret the epistemic modality through a counterpart semantics based on the relativized indistinguishability of section 3. Assume an interpreted system $\mathcal{I}$, and an assignment $V : \mathit{VAR} \longrightarrow \mathcal{T}_{\equiv}$. Assignments are extended homomorphically to terms in the usual way, and $V[x \mapsto M]$ is V except that x is assigned M.

Definition 4 (Truth)

$$\begin{aligned}
s, V \models_{\mathcal{I}} \Box_A F &\Leftrightarrow \forall s' \in S : \forall \rho : \\
&\quad\ s \sim_A^{\rho} s' \Rightarrow s', \rho \circ V \models_{\mathcal{I}} F \\
s, V \models_{\mathcal{I}} t = t' &\Leftrightarrow V(t) = V(t') \\
s, V \models_{\mathcal{I}} p(t_1, ..., t_n) &\Leftrightarrow \langle V(t_1), ..., V(t_n) \rangle \in I(p, s) \\
s, V \models_{\mathcal{I}} \forall x.F &\Leftrightarrow \forall M \in \mathcal{T}_{\equiv} : s, V[x \mapsto M] \models_{\mathcal{I}} F \\
s, V \models_{\mathcal{I}} \forall m.F[m/x] &\Leftrightarrow \forall M \in \mathcal{T} : s, V \models_{\mathcal{I}} F[M/x]
\end{aligned}$$

For Boolean operators we assume standard truth conditions. The semantics for the modality follows counterpart semantics in that it checks F at s' with respect to the corresponding assignment $\rho \circ V$ instead of the original assignment V, as in basic Kripke semantics. As a result, cryptographic omniscience (1) fails.

Validity is defined as usual: A statement F is valid in interpreted system $\mathcal{I}$, written $\models_{\mathcal{I}} F$, if for all $s \in S$ and all assignments V, we have $s, V \models_{\mathcal{I}} F$. Statement F is valid in system $\mathcal{S}$, written $\models_{\mathcal{S}} F$, if F is valid in all interpreted systems based on $\mathcal{S}$. Statement F is valid, in symbols $\models F$, if F is valid in all systems. Statement F is valid at a state s, written $s \models F$, if $s, V \models_{\mathcal{I}} F$ for all assignments V and all interpreted systems $\mathcal{I}$ containing s.

Example 3 *Consider the interpreted system $\mathcal{I}$ from example 2. Since sender $\notin$ LOC$|C$, C does not know the sender:* $\models_{\mathcal{I}} \forall x.(@_{\mathit{sender}}\, x \rightarrow \neg\Box_C @_{\mathit{sender}}\, x)$. *However, since post $\in$ LOC$|C$, C knows (as "bitstring") what message is posted:* $\models_{\mathcal{I}} \forall x.(@_{\mathit{post}}\, x \rightarrow \Box_C @_{\mathit{post}}\, x)$. *On the other hand, C need not know the structure of the posted message:* $\not\models_{\mathcal{I}} \forall m.(@_{\mathit{post}}\, m \rightarrow \Box_C @_{\mathit{post}}\, m)$. *From the former validity and the latter invalidity, it follows that cryptographic omniscience (1) fails:* $\not\models_{\mathcal{I}} x = M \rightarrow \Box_C x = M$.

In the following theorem, assume a hash operator h and assume that, for each $s \in S$, there are at least two messages that A cannot infer at s, i.e., $|\overline{\mathit{Inferable}(A, s)}| \geq 2$.

Theorem 2 (Characterization of Inference)
$\models A \mathit{\ infers\ } x \leftrightarrow \exists y.\Box_A y = h(x)$

In theorem 2, recall that the interpretation of predicate $A \mathit{\ infers}$ at state s is $\mathit{Inferable}(A, s)$. In light of theorem 2, we introduce $\Box_A\, x$, read "A knows x", as an abbreviation for the statement $\exists y.\Box_A y = h(x)$. We write $\Box_A\, \overline{x}$ for $\bigwedge_i \Box_A\, x_i$, and we write $\neg\Box_A\, \overline{x}$ for $\bigwedge_i \neg\Box_A\, x_i$.

In the following theorem, we assume local states $s|A$ and $s'|A$ are normal. Moreover, we assume predicates $\mathcal{P}$ includes $@_l$, for $l \in LOC$.

Theorem 3 (Logical Characterization of $\approx$) *The following are equivalent:*

1. $s|A \approx s'|A$.
2. $s \models \Box_A F$ *iff* $s' \models \Box_A F$, *for all statements* F.

6 Security Protocol Examples

Mix Consider a mix in the style of [11]. The mix inputs a sequence of encryptions $enc(M_1, pk(K), N_1)$,..., $enc(M_l, pk(K), N_l)$, where $pk(K)$ is the mix's public key, generated by a secret K, and N_i is a random seed. The mix later outputs the encryption content in random order: $M_{\pi(1)}, ..., M_{\pi(l)}$, for some random permutation π on $\{1...l\}$. A spy should not be able to link inputs to outputs:

$$mix\ inputs\ x \wedge mix\ outputs\ y \rightarrow \neg\Box_{spy} x\ contains\ y \quad (2)$$

$$mix\ inputs\ x \wedge mix\ outputs\ y \rightarrow \Diamond_{spy} x\ contains\ y \quad (3)$$

where $x\ contains\ y$ abbreviates $\exists z.\exists z'.x = enc(y, z, z')$, and $\Diamond_{spy}$ abbreviates $\neg\Box_{spy}\neg$. Perhaps, our concern is that the mix detects, rather than prevents, information leakage, i..e, whenever the spy determines a link, the mix knows this:

$$mix\ inputs\ x \wedge mix\ outputs\ y \rightarrow$$
$$\Box_{spy} x\ contains\ y \rightarrow \Box_{mix}\Box_{spy} x\ contains\ y$$

In [14], we check the above security goals in some interpreted systems which implement the protocol. As expected, specifications (2) and (3) fail if the implementation allows input replays, or if it allows inputs of various lengths. By adding a length-computing operator len, with equations such as $len(enc(M, K, N)) \equiv len(M)$, the spy is given the ability to perform length-comparisons.

Crowds We consider a Crowds-style protocol [29], which allows members of a crowd to communicate without non-crowd members knowing who is talking to whom. The agents of a set $Crowd$ share a symmetric key K. Crowd member A sends a message M anonymously to crowd member B, by sending the symmetric encryption $enc(pair(B, M), K)$ to some random crowd member C_1, who in turn sends this encryption to B or to a random forwarder C_2, and so on until the encryption reaches B. In addition to crowd members, there are local spies, who observe and control the traffic in some part of the network. Receiver anonymity means that a spy cannot tell the intended destination of a given message x:

$$x\ is\ for\ A \rightarrow \Diamond_{spy} x\ is\ for\ B$$

for crowd members A and B outside the observation domain of spy. Since spies can block messages, $x\ is\ for\ A$ must be defined in terms of x's structure, and not in terms of where x eventually ends up: $x\ is\ for\ A$ might abbreviate $\exists y.fst(dec(x, y)) = A \wedge \bigvee_B B\ sent\ x$. Sender anonymity means that a spy cannot tell the originator of a message:

$$A\ originated\ x \rightarrow \Diamond_{spy} B\ originated\ x$$

for crowd members A and B who are outside the reach of spy. Although it does not specify knowledge of structure, this specification relies on the absence of (1), since it specifies what the spy knows of an undecrypted message [14].

Dual Signature Consider a purchasing protocol involving three parties, a customer C, a merchant M, and a bank B. To order an item x_i using payment data (credit card number, etc.) x_p, the customer produces a dual signature [26] using the private signing key x_s:

$$dual(x_i, x_p, x_s) =_{df} sign(pair(h(x_i), h(x_p)), x_s)$$

The merchant M receives $dual(x_i, x_p, x_s)$, x_i and $h(x_p)$, while B receives $dual(x_i, x_p, x_s)$, $h(x_i)$ and x_p. The dual signature hides the order item x_i from B, and the payment data x_p from M, nonetheless the dual signature links x_i to x_p so that their correspondence cannot later be disputed. We now consider in more detail what B learns during protocol execution. Let variable $x_d = dual(x_i, x_p, x_s)$ refer to the dual signature that C creates in the current run. At the end of the protocol, B knows that the dual signature was produced by C's private signing key:

$$\Box_B C\ signed\ x_d$$

where $C\ signed\ x_d$ might abbreviate $\exists x_s.\exists y.x_d = sign(y, x_s) \wedge x_s\ sign\ key\ of\ C$. Using $h(x_i)$ and x_p, the bank can determine the payment data x_p inside:

$$\Box_B x_d\ contains\ payment\ x_p$$

where $x_d\ contains\ payment\ x_p$ abbreviates $\exists x_i.\exists x_s.x_d = dual(x_i, x_p, x_s)$. But, B cannot determine the order item:

$$\neg\Box_B x_d\ contains\ item\ x_i$$

where $x_d\ contains\ item\ x_i$ abbreviates $\exists x_p.\exists x_s.x_d = dual(x_i, x_p, x_s)$. Finally, B is assured that M can determine the order item:

$$\Box_B \exists x_i.\Box_M x_d\ contains\ item\ x_i$$

7 Axiomatization

In table 1, we define a Hilbert-style axiomatization, relative to a message congruence $\equiv$ with a hash operator h.

First-order

(Ins x) $\forall x.F \rightarrow F[y/x]$

(Bound x) $\forall x.F \leftrightarrow F$, if x is not free in F

(Dist x) $\forall x.(F \rightarrow F') \rightarrow \forall x.F \rightarrow \forall x.F'$

(Subst) $t = t' \rightarrow F[t/x] \rightarrow F[t'/x]$, if F has no modality

(Eq) $t = t$

(Taut) F, if F is truth functional tautology

(Gen x) $\dfrac{F}{\forall x.F}$

(Ins m) $\forall m.F[m/x] \rightarrow F[M/x]$

(Bound m) $\forall m.F[m/x] \leftrightarrow F$, if x is not free in F

(Dist m) $\forall m.(F[m/x] \rightarrow F'[m/x]) \rightarrow \forall m.F[m/x] \rightarrow \forall m.F'[m/x]$

(Ins t) $\forall x.F \rightarrow F[t/x]$, if F has no modality

($m\,x$) $\exists m.x = m$

(MP) $\dfrac{F \rightarrow F',\ F}{F'}$

Modal S5

(K) $\Box_A(F \rightarrow F') \rightarrow (\Box_A F \rightarrow \Box_A F')$

(T) $\Box_A F \rightarrow F$

(4) $\Box_A F \rightarrow \Box_A \Box_A F$

(5) $\neg\Box_A F \rightarrow \Box_A \neg\Box_A F$

(Nec) $\dfrac{F}{\Box_A F}$

Knowledge and Cryptography

($\Box$1) $\Box_A\,\overline{x} \rightarrow (y = f(\overline{x}) \rightarrow \Box_A y = f(\overline{x}))$

($\Box$2) $x = y \rightarrow \Box_A x = y$

($\Box$3) $y = f(\overline{x}) \rightarrow \Box_A\,\overline{x} \rightarrow \Box_A\,y$

($\Box$4) $\Box_A F(\overline{x},\overline{y}) \rightarrow \Box_A \overline{y} \rightarrow \neg\Box_A \overline{x},\overline{z}$
$\rightarrow \bigwedge_{i,j}(x_i = x_j \leftrightarrow z_i = z_j) \rightarrow F[\overline{z}/\overline{x}]$

($\Box$5) $\exists x.\exists y.x \neq y \wedge \neg\Box_A\,x, y$

Omega

($\equiv$) $M = M'$, if $M \equiv M'$

($\not\equiv$) $M \neq M'$, if $M \not\equiv M'$

(Gen m) $\dfrac{F[M/x],\ \text{all } M \in \mathcal{T}}{\forall m.F[m/x]}$

Figure 1. Axioms and Rules

The first group of axioms and rules is inherited from first-order logic, and includes a (less standard) axiom connecting the two kinds of quantifier. The second group is modal S5, as expected for introspective knowledge. The third group contains five axioms for the interaction between knowledge and cryptography. While axiom ($\Box$2) is well-known from first-order modal logic, the other four axioms are novel. Axiom ($\Box$1) reflects the assumption that each operator f is feasible to compute. Axiom ($\Box$3) states that inferred messages are closed under operators f. Axiom ($\Box$4) reflects the assumption that non-inferred values are "anonymous": The statement says that the agent knows a property of some non-inferred values $\overline{x}$ only if this property holds of any non-inferred valules $\overline{z}$ with the same pattern of identities. More precisely, assume $\overline{x},\overline{y}$ are all variables free in F. Assume A infers messages $\overline{y}$ but A cannot infer any of messages $\overline{x},\overline{z}$. Assume, finally, that $\overline{x}$ and $\overline{z}$ have the same pattern of identities. Then, $\Box_A F \rightarrow F[\overline{z}/\overline{x}]$. Axiom ($\Box$5) reflects the restriction on systems needed for theorem 2, namely that there are at least two messages that agent A does not infer. In [14], we provide correspondence results for axioms ($\Box$1) and ($\Box$4). The fourth group includes all equalities and inequalities from $\equiv$ and an omega-rule for the *de dicto* quantifier. Write $\vdash F$ when F is a derivable theorem. We obtain some standard theorems, such as Barcan and converse Barcan for both kinds of quantifier, and unrestricted substitution for variables. Some characteristic theorems used in our results are: $\Box_A\,x \rightarrow \Box_A\Box_A\,x$, $\neg\Box_A\,x \rightarrow \Box_A\neg\Box_A\,x$, and if t contains no secret constant c then $x = t \rightarrow \Box_A\,\mathit{VAR}(t) \rightarrow \Box_A x = t$.

8 Soundness and Completeness

We arrive at the main result. We consider only systems where $|\overline{\mathit{Inferable}(A,s)}| \geq 2$, for all $s \in S$ and all $A \in \mathcal{A}$.

Theorem 4 $\vdash F \Leftrightarrow \models F$

In the rest of this section, we build the completeness construction. The following sections - sections 9 and 10 - can be read independently. The completeness construction uses abstract counterpart models, with arbitrary states ("possible worlds") w, arbitrary domain of quantification, arbitrary accessibility relation $\longrightarrow_A^{\rho}$ and arbitrary (non-rigid) interpretation of function symbols. The first step is a standard canonical Kripke model $\mathcal{K}$, which is transformed into a counterpart model $\mathcal{K}^\star$ by adding some epistemic transitions. For each transition $w \longrightarrow_A w'$ in $\mathcal{K}$, a transition $w \longrightarrow_A^{\pi} w'$ is added, where π is any permutation of non-inferred items at w, i.e., items satisfying $\neg\Box_A\,x$ at w. Continuing, we define a morphism d, which morphs $\mathcal{K}^\star$ into a counterpart model $\mathsf{d}(\mathcal{K}^\star)$ with a rigid interpretation of functions symbols f, given by the background message equivalence $\equiv$. Finally, a morphism w transforms $\mathsf{d}(\mathcal{K}^\star)$ into a counterpart

model $\mathsf{w}(\mathsf{d}(\mathcal{K}^\star))$, which is equivalent to an interpreted system.

Abstract Counterpart Model We review some basics from (a variant of) counterpart semantics (cf. [18]). An abstract counterpart model is a structure $\mathcal{C} = \langle W, D, \longrightarrow, I\rangle$, defined as follows. W is a non-empty set of worlds w, and D is a non-empty domain of objects d. For $A \in \mathcal{A}$, $\rightarrow_A \subseteq W \times (D \longrightarrow D) \times W$ is the epistemic accessibility relation. Informally, $w \longrightarrow_A^\rho w'$ means that w and w' are indistinguishable for A and each $d \in D$ at w corresponds for A to $\rho(d)$ at w'. I is a world-relative interpretation, i.e., $I(c, w)$ is a member of D, $I(f, w)$ is an operation in D matching the arity of f, and $I(p, w)$ is a relation in D matching the arity of p. Thus, the interpretation of f and c is left open, and need not be rigid. An assignment in $\mathcal{C}$ is a function $V : \mathit{VAR} \longrightarrow D$. Assignments are extended to arbitrary terms with respect to a world w as usual: $V(x, w) = V(x)$, $V(c, w) = I(c, w)$, $V(f(t_1, ..., t_n), w) = I(f, w)(V(t_1, w), ..., V(t_n, w))$. Truth conditions are as expected:

$$
\begin{array}{rcl}
w, V \models_{\mathcal{C}} \Box_A F & \Leftrightarrow & \forall w' \in W : \forall \rho : w \longrightarrow_A^\rho w' \Rightarrow \\
 & & w', \rho \circ V \models_{\mathcal{C}} F \\
w, V \models_{\mathcal{C}} t = t' & \Leftrightarrow & V(t, w) = V(t', w) \\
w, V \models_{\mathcal{C}} p(t_1, ..., t_n) & \Leftrightarrow & \langle V(t_1, w), ..., V(t_n, w)\rangle \in I(p, w) \\
w, V \models_{\mathcal{C}} \forall x.F & \Leftrightarrow & \forall d \in D : w, V[x \mapsto d] \models_{\mathcal{C}} F \\
w, V \models_{\mathcal{C}} \forall m.F[m/x] & \Leftrightarrow & \forall M \in \mathcal{T} : w, V \models_{\mathcal{C}} F[M/x]
\end{array}
$$

Any interpreted system $\mathcal{I} = \langle \mathit{LOC}, S, |, I\rangle$ determines a counterpart model $\mathcal{C}_\mathcal{I} = \langle S, \mathcal{T}_\equiv, \sim, I'\rangle$, where $\sim_A$ is defined as in section 4 and $I'(p, s) = I(p, s)$ and $I'(f, w) = f$ and $I'(c, w) = c$. We say that $\mathcal{C}_\mathcal{I}$ is induced by $\mathcal{I}$.

Corollary 3 *$s, V \models_\mathcal{I} F$ iff $s, V \models_{\mathcal{C}_I} F$.*

A counterpart model $\mathcal{C}$ is Kripkean if $w \longrightarrow_A^\rho w'$ implies that $\rho = Id$, where Id is identity on D. When $\mathcal{C}$ is Kripkean, we omit the index Id, and write $w \longrightarrow_A w'$ for the transition $w \longrightarrow_A^{Id} w'$. We say that substitutions are bijective in $\mathcal{C}$, if $w \longrightarrow_A^\rho w'$ implies ρ is a permutation on D.

Assume a counterpart model $\mathcal{C} = \langle W, D, \longrightarrow, I\rangle$. Assume a set W' of worlds and a domain D'. A morphism from $\mathcal{C}$ to W' and D' is a pair w, d such that:

- $\mathsf{w} : W \longrightarrow W'$ is a bijective map
- $\mathsf{d}_w : D \longrightarrow D'$ is a bijective map, for each $w \in W$

The morphism w, d is a domain-morphism, if $W = W'$ and w is identity on W. The morphism w, d is a world-morphism, if $D = D'$ and d_w is identity on D. For domain-morphisms, we leave the identity w implicit. Similarly, for world-morphisms, we leave the mapping d implicit. Let w, d be a morphism from $\mathcal{C}$ to W' and D'. The application of w, d on $\mathcal{C}$ is $\mathsf{wd}(\mathcal{C}) = \langle W', D' \xrightarrow{\mathsf{wd}}, I^{\mathsf{wd}}\rangle$, where

- $\mathsf{w}(w) \xrightarrow{\mathsf{wd}}_A^\rho \mathsf{w}(w')$ iff $w \longrightarrow_A^{\rho'} w'$ where $\rho' = \mathsf{d}_{w'}^{-1} \circ \rho \circ \mathsf{d}_w$.
- $I^{\mathsf{wd}}(o, \mathsf{w}(w)) = \mathsf{d}_w(I(o, w))$, $o \in \mathit{SEC} \cup \Sigma \cup \mathcal{P}$.

Thus, $\mathsf{wd}(\mathcal{C})$ is the result of pointwise "relabelling" $\mathcal{C}$ through w and d.

Lemma 3 *$w, V \models_\mathcal{C} F \Leftrightarrow \mathsf{w}(w), \mathsf{d}_w \circ V \models_{\mathsf{wd}(\mathcal{C})} F$.*

Canonical Kripke Model The canonical Kripke model is built on saturated sets in the usual way [19]. A statement F is derivable from a set Γ of statements, in symbols $\Gamma \vdash F$, if there is a finite number of statements $F_1, ..., F_n \in \Gamma$ such that $\vdash F_1, ..., F_n \rightarrow F$. The set Γ is consistent if $\Gamma \nvdash \bot$, and Γ is maximal consistent if it is consistent and no larger set is consistent. The set Γ is omega-complete if whenever $\Gamma \vdash F[y/x]$ for all $y \in \mathit{VAR}$ then $\Gamma \vdash \forall x.F$ and, also, whenever $\Gamma \vdash F[M/x]$ for all $M \in \mathcal{T}$ then $\Gamma \vdash \forall m.F[m/x]$. The set Γ is saturated if it is maximal consistent and omega-complete.

Given a saturated set w_0, the canonical Kripke model $\mathcal{K} = \langle W, D, \rightarrow, I\rangle$ is defined as follows. The set W of worlds is the set of all saturated sets which contain $x = y$ if and only if w_0 contains $x = y$. The domain D is the set of equivalence classes $|x| = \{y : x = y \in w_0\}$. The epistemic accessibility is given by: $w \longrightarrow_A w' \Leftrightarrow w|A \subseteq w'$, where $w|A$ is $\{F : \Box_A F \in w\}$. Finally, the interpretation is defined as follows: $I(f, w)(|x_1|, ..., |x_n|) = |y|$ iff $(f(x_1, ..., x_n) = y) \in w$, and $I(c, w) = |y|$ iff $(c = y) \in w$. The canonical assignment $V_\mathcal{K}$ assigns $|x|$ to variable x.

Lemma 4 (Truth Lemma for $\mathcal{K}$) *$w, V_\mathcal{K} \models_\mathcal{K} F \Leftrightarrow F \in w$*

Anonymous Non-inferred Items We transform $\mathcal{K}$ into a model where non-inferred items, i.e., items satisfying $\neg\Box_A x$, are anonymous in the sense that every permutation of such items is "epistemically possible". The transformation relies on axiom ($\Box 4$). Assume a counterpart model $\mathcal{C} = \langle W, D \longrightarrow, I\rangle$. Write $\mathit{Inferable}_\mathcal{C}(A, w)$ for the set of items inferred by agent A at world w, i.e, $\mathit{Inferable}_\mathcal{C}(A, w)$ is $\{d \in D \mid w, V[\mapsto d] \models_\mathcal{C} \Box_A x\}$. The anonymization of $\mathcal{C}$ is the model $\mathcal{C}^\star = \langle W, D \xrightarrow{\star}, I\rangle$, where $\xrightarrow{\star}$ is the least extension of $\longrightarrow$ such that

$$w \xrightarrow{\star}_A^\rho w' \Rightarrow w \xrightarrow{\star}_A^{\rho\circ\pi} w'$$

for every permutation π on $\overline{\mathit{Inferable}_\mathcal{C}(A, w)}$. ($\pi$ is extended to the whole domain D in the expected way: $\pi(d) = d$ if $d \in \mathit{Inferable}_\mathcal{C}(A, w)$.)

Lemma 5 *$w, V \models_\mathcal{K} F \Leftrightarrow w, V \models_{\mathcal{K}^\star} F$.*

Rigid Operators We define a domain-morphism d, which morphs $\mathcal{K}^\star$ into a model $\mathsf{d}(\mathcal{K}^\star)$ where operators f and constants c have their intended, rigid denotation, given by the background equivalence $\equiv$. The transformation relies on axioms ($m\,x$), ($\equiv$) and ($\not\equiv$). For each $w \in W$, we relate D and $\mathcal{T}_\equiv$ by the relation: $\mathsf{d}_w = \{\langle |x|, M\rangle \mid x = M \in w\}$.

Lemma 6 *d is a morpism from $\mathcal{K}^\star$ to W and $\mathcal{T}_\equiv$.*

Let $\mathsf{d}(\mathcal{K}^\star) = \langle W, \mathcal{T}_\equiv, \xrightarrow{\mathsf{d}}, I^{\mathsf{d}}\rangle$ be application of d on $\mathcal{K}^\star$.

Lemma 7 *$I^{\mathsf{d}}(f, w) = f$ and $I^{\mathsf{d}}(c, w) = c$.*

Canonical Interpreted System Finally, we define a world-morphism w, which morphs $\mathsf{d}(\mathcal{K}^\star)$ into a model $\mathsf{w}(\mathsf{d}(\mathcal{K}^\star))$ induced by an interpreted system. The transformation step relies on axioms ($\Box 1$) and ($\Box 5$). We assume the following set of store locations: $\mathit{LOC} = \mathcal{F} \cup ((D \cup \mathcal{F}) \times \mathcal{A})$ (where D is the domain in $\mathcal{K}$ and $\mathcal{K}^\star$). Each agent observes store locations indexed by itself: $\mathit{LOC}|A = (D \cup \mathcal{F}) \times \{A\}$. The morphism w maps W to states over *LOC* defined by:

1. $\mathsf{w}(w)(\langle |x|, A\rangle) = \mathsf{d}_w(|x|)$, if $|x| \in \mathit{Inferable}_{\mathcal{K}}(w, A)$.
2. $\mathsf{w}(w)(\langle |x|, A\rangle) = \bot$, if $|x| \notin \mathit{Inferable}_{\mathcal{K}}(w, A)$.
3. $\mathsf{w}(w)(\langle F, A\rangle) = \top$, if $\Box_A\, F \in w$.
4. $\mathsf{w}(w)(\langle F, A\rangle) = \bot$, if $\Box_A\, F \notin w$.
5. $\mathsf{w}(w)(F) = \top$, if $F \in w$.
6. $\mathsf{w}(w)(F) = \bot$, if $F \notin w$.

where $\bot$ and $\top$ are two non-equivalent 0-arity operators from Σ. (If there is only one such operator, i.e., the single agent A, then let $\bot = A$ and $\top = h(A)$.) Requirements (3) and (4) on w encode the knowledge state $w|A$ inside the local state $\mathsf{w}(w)|A$. Requirements (5) and (6) ensure injectivity. Requirements (1) and (2), together with (3) and (4), ensure that the same permutations ρ are possible between $\mathsf{w}(w)$ and $\mathsf{w}(w')$ in $\sim_A$ as between w and w' in $\xrightarrow{\mathsf{d}}_A$:

Lemma 8 *$w \xrightarrow{\mathsf{d}}{}^{\rho}_{A} w'$, if and only if, $\mathsf{w}(w) \sim^{\rho}_{A} \mathsf{w}(w')$.*

Let the canonical interpreted system be $\mathcal{I} = \langle \mathit{LOC}, S, |, I\rangle$, where $S = \{\mathsf{w}(w) : w \in W\}$ and $I(p, \mathsf{w}(w)) = \{\langle M_1, ..., M_n\rangle \mid \mathsf{w}(w)(p(M_1, ..., M_n)) = \top\}$.

Lemma 9 *$\mathsf{w}(\mathsf{d}(\mathcal{K}^\star))$ is induced by $\mathcal{I}$.*

By axiom ($\Box 5$), $\mathcal{I}$ satisfies our restriction on systems.

Theorem 5 *Every consistent statement is satisfiable in some interpreted system.*

From theorem 5, we get completeness theorem 4.

9 Embedding of BAN and SVO

The BAN Modality We show how to capture the epistemic modality from BAN logic [10]. BAN statements β are propositional, built from closed atomic statements $p(M_1, ..., M_n)$, epistemic modalities and Boolean operators. (Original BAN includes "idealized" messages, but no negation. But, these differences are othogonal the our concern here.) Necessitation (Nec) is incompatible with BAN, since terms M refers *de re* [12, 13, 14]. We propose that the following omega-weakening of necessitation is faithful to BAN:

$$\frac{\beta[\overline{M}/\overline{c}],\ \text{all}\ \overline{M}}{\Box_A\, \overline{M} \rightarrow \Box_A \beta[\overline{M}/\overline{c}]} \quad (\textit{WNec})$$

where $\overline{c}$ lists all constant occurring in β. For instance:

$$\frac{\mathit{enc}(M, K)\ \mathit{contains}\ M,\ \text{all}\ M, K}{\Box_A\, M, K \rightarrow \Box_A\, \mathit{enc}(M, K)\ \mathit{contains}\ M}$$

In [14], we use rule *WNec* to derive BANs message meaning rule and another characteristic BAN rule. Similar weakenings of necessitation appear in [12, 13, 23]. Define translation τ from BAN into our logic:

$$\beta(\overline{M})^\tau = \exists \overline{x}.(\beta(\overline{x}) \wedge \bigwedge_i x_i = M_i)$$

where $\overline{M}$ is a list $\langle M_1, ..., M_n\rangle$ of all ground terms occurring as arguments to predicates in β, $\beta(\overline{x})$ is the result of substituting x_i for M_i, and $\exists\overline{x}$ abbreviates $\exists x_1 ... \exists x_n$. For instance, τ translates $\Box_A \Box_B A\ \mathit{receives}\ \mathit{enc}(M, K)$ to $\exists x.x = \mathit{enc}(M, K) \wedge \Box_A \Box_B A\ \mathit{receives}\ x$. Let WNec^τ be the τ-translations of *WNec*.

Proposition 2 *WNec^τ is a derived rule.*

The embedding τ induces a new truth condition:

Proposition 3 *The following are equivalent:*

- $s \models_{\mathcal{I}} (\Box_A \beta(\overline{M}))^\tau$
- $\forall s' \in S : \forall \rho : s \sim^{\rho}_{A} s' \Rightarrow s' \models_{\mathcal{I}} \beta(\rho(\overline{M}))^\tau$

Providing a faithful semantics for BAN's modality has been a longstanding problem. The truth condition in proposition 3 is, essentially, a generalization to an arbitrary equational theory of the semantics proposed for BAN in [12, 13].

The SVO Modality Protocol derivations in SVO [32], a successor to BAN, uses variables (represented as stars: $\star$, $\star_x$, $\star_y$, etc.) to refer *de re* to possibly undecrypted content. The derivations assume that seeing implies knowledge of

seeing to the extent that the seen message can be decrypted. For instance, for the equational theory in example 1,

$$A \textit{ sees } enc(\textit{pair}(x, x'), \textit{pk}(z)),\ A \textit{ infers } z \rightarrow \\ \Box_A A \textit{ sees } enc(\textit{pair}(x, x'), \textit{pk}(z)) \qquad (4)$$

Seeing introspection principles, such as (4), are not justified by the proof system in [32], but the authors remark that it would be straightforward to capture such implications in an axiom. We propose the following axiom:

$$A \textit{ sees } T \rightarrow \Box_A \textit{ VAR}(T) \rightarrow \Box_A A \textit{ sees } T \quad (SEE)$$

where T is any term without constants from *SEC*. The semantics in [32] does not support (4) or SEE. More generally, the semantics there does not support *de re* reference of variables. We show, however, that our semantics fits (4) and *SEE*. Let *SVO* be the following assumption:

$$\forall x.(A \textit{ sees } x \rightarrow \Box_A A \textit{ sees } x) \wedge \exists x.(\neg \Box_A x \wedge \neg A \textit{ sees } x)$$

Trivially, an interpreted system $\mathcal{I}$ satisfies the first conjunct of *SVO* if, and only if, $\rho(I(A \textit{ sees}, s)) \subseteq I(A \textit{ sees}, s')$ whenever $s \longrightarrow_A^{\rho} s'$ in $\mathcal{I}$.

Proposition 4 $\vdash$ *SVO* $\rightarrow$ *SEE and* $\vdash$ *SVO* $\rightarrow (4)$.

10 Related Work

In [13] we use a propositional variant of the semantics presented here to account for BAN, and [12] gives a completeness result. For the relationship to [13], see proposition 3 and its explanation. The completeness result of this paper is stronger and much less ad hoc: The logic is richer, we avoid restriction to finite message spaces (which does not square well with most formal accounts of cryptography), we avoid the ad hoc "internal actions" used in [12] with no clear computational meaning, and the axiomatization is much less schematic and does not rely on a specific term algebra.

In the security literature, there are several semantics for epistemic logic. Often, the "standard" multi-agent system semantics [16] is used, with identity as the equivalence on local states (cf [21, 30]). But in connection with cryptography, identity is clearly inappropriate as equivalence. This is manifested in counter-intuitive validities such as $\exists x.A \textit{ receives } enc(M, x) \rightarrow \Box_A \exists x.A \textit{ receives } enc(M, x)$. Our semantics is most closely related to the AT-semantics [4, 5]. The AT semantics, which applies only to symmetric encryption, reduces indistinguishability to identity on expressions that use a fixed symbol $\Box$ to represent undecryptable sub-expressions, and thereby loses all information about undecryptable data, including knowledge that may have been obtained indirectly, for instance, as a result of the protocol being executed (cf. the examples in section 6, also the unsoundness of BAN's message meaning rule in AT). We emphasize that thare are no completeness results for AT-semantics, or its variants (cf. [32]). Moreover, since AT-semantics, and its variants, follow basic Kripke semantics, they render agents cryptographically omniscient (1). On the other hand, approaches based on explicit knowledge avoid cryptographical omniscience but have other drawbacks, discussed briefly in section 1. The compositional protocol logic of Durgin et al [15] uses knowledge only in terms of Dolev-Yao type message deduction. The role of permutations in our semantics is slightly reminiscent of [8, 33]. A version of theorem 3 for an AT-style semantics was proposed by S. Kramer (private correspondence).

The application of epistemic logic to cryptographic protocol analysis goes back to BAN logic [10]. Our protocol examples are, more directly, inspired by anonymity specifications in [21] and specifications for the SET protocol in [7]. We refer to [23] for a comprehensive dictionary of epistemic security specifications.

Interaction axiom ($\Box 3$), and, to a lesser degree, interaction axiom ($\Box 1$), are reminiscent of BAN-style proof systems. However, BAN-style proof systems contain only ad hoc rules specific to concrete predicates.

11 Concluding Remarks

One issue left open by our work is the role of the *de dicto* quantifier $\forall m$. We have been unable to obtain completeness for a compact logic which does not use this quantifier. A candidate omega-rule is:

$$\frac{x = M \rightarrow F,\ \text{all } M \in \mathcal{T}}{\forall x.F}\ .$$

We can show that the *de dicto* quantifier adds to the expressive power. Let $\Sigma = \mathcal{A} = \{A\}$, i.e., let there be only one public operator, namely the agent identifier A, and let $\equiv$ be identity on ground terms.

Proposition 5 *No statement free of the* de dicto *quantifier is equivalent to* $\exists m.\exists x.x \neq A \wedge \Box_A x = m$.

Our semantics is formulated in a counterpart semantics framework, although the choice of framework is, to some extent, a matter of taste. It is possible to reformulate the semantics in the framework of first-order intensional logic [9]. In such a framework, variables denote intensions, i.e., functions from states to individuals. In our setting, individuals are messages, and intensions are terms built from store locations and operators, such as the s-terms of section 2. However, reformulating our logic as a first-order intensional logic would, it seems, make security specifications more complex. A statement $\Box_A F(x)$ in our logic translates to, it seems, something like $\exists y.x = y \wedge A\textit{-term}\,(y) \wedge \Box_A F(y)$,

where *A-term* is a predicate which applies to an intension if that intension is built from feasibly computable operators and store locations A can observe. An additional intension y is needed, since the intension x might be built from store locations not observed by A. As a result, the translation induces extra nesting of quantifiers and modalities. To illustrate, the statement $\Box_B \Box_A F(x)$ translates to $\exists y.x = y \wedge$ *B-term*$(y) \wedge \Box_B \exists z.z = y \wedge$ *A-term*$(z) \wedge \Box_A F(z)$.

The proposed logic is static only: It expresses properties of states, but not of computations. In the future, we plan to extend the completeness result to include temporal modalities, and to link to concepts in information flow security, and to behavioral equivalences for applied pi.

Acknowledgements

We are grateful to Karl Meinke for helpful discussions and to Katie Asplund Cohen for extensive feedback on previous drafts. This work was partially supported by Swedish Research Council grants 2003-6108 and 2003-2597.

References

[1] M. Abadi, M. Baudet, and B. Warinschi. Guessing attacks and the computational soundness of static equivalence. In *FoSSaCS'06*, pages 398–412, 2006.

[2] M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. *Theor. Comput. Sci.*, 367(1-2):2–32, 2006.

[3] M. Abadi and A. D. Gordon. A bisimulation method for cryptographic protocols. *Nordic J. of Computing*, 5(4), 1998.

[4] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

[5] M. Abadi and M. Tuttle. A semantics for a logic of authentication. In *PODC'91*, pages 201–216, 1991.

[6] R. Accorsi, D. A. Basin, and L. Viganò. Towards an awareness-based semantics for security protocol analysis. *Electr. Notes Theor. Comput. Sci.*, 55(1), 2001.

[7] N. Agray, W. van der Hoek, and E. P. de Vink. On ban logics for industrial security protocols. In B. Dunin-Keplicz and E. Nawarecki, editors, *From Theory to Practice in Multi-Agent Systems, Second International Workshop of Central and Eastern Europe on Multi-Agent Systems (CEEMAS 2001)*, volume 2296 of *Lecture Notes in Computer Science*, pages 29–36. Springer, 2001.

[8] P. Bieber. A logic of communication in hostile environments. In *Third IEEE Computer Security Foundations Workshop (CSFW'90)*, pages 14–22. IEEE Computer Society Press, 1990.

[9] T. Brauner and S. Ghilardi. First-order modal logic. In F. W. Patrick Blackburn, Johan van Benthem, editor, *Handbook of Modal Logic: Volume III*. Elsevier, 2006.

[10] M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. *ACM Trans. Comput. Syst.*, 8(1):18–36, 1990.

[11] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.

[12] M. Cohen and M. Dam. A completeness result for BAN logic. In *2005 International Workshop on Methods for Modalities (M4M-05)*, pages 202–219, 2005.

[13] M. Cohen and M. Dam. Logical omniscience in the semantics of BAN logic. In *Foundations of Computer Security (FCS'05)*, pages 121–132, 2005.

[14] M. Cohen and M. Dam. A complete axiomatization of knowledge and cryptography. Research report TRITA-CSC-TCS-2007:1, KTH CSC, 2007. http://web.it.kth.se/∽mfd/TRITACSCTCS1.pdf.

[15] N. Durgin, J. Mitchell, and D. Pavlovic. A compositional logic for proving security properties of protocols. *J. Comput. Secur.*, 11(4):677–721, 2004.

[16] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[17] C. Fournet and M. Abadi. Mobile values, new names, and secure communication. In *The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, ACM SIGPLAN Notices 36(3)*, pages 104–115, 2001.

[18] D. Gabbay, V. Shehtman, and D. Skvortsov. Quantification in nonclassical logic. 2006. Manuscript.

[19] J. W. Garson. Quantification in modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume II*. Reidel, 1984.

[20] O. Goldreich. *Foundations of Cryptography*, volume Basic Tools. Cambridge University Press, 2001.

[21] J. Halpern and K. O'Neill. Anonymity and information hiding in multiagent systems. In *Proc. CSFW'03*, pages 75–88, 2003.

[22] J. Y. Halpern and R. Pucella. Modeling adversaries in a logic for security protocol analysis. In *Proc. FASec*, pages 115–132, 2002.

[23] S. Kramer. Logical concepts in cryptography. Cryptology ePrint Archive, Report 2006/262, 2006.

[24] D. Lewis. Counterpart theory and quantified modal logic. *Journal of Philosophy*, 65:113–126, 1968.

[25] A. Lomuscio and B. Wozna. A combination of explicit and deductive knowledge with branching time: Completeness and decidability results. In M. Baldoni, U. Endriss, A. Omicini, and P. Torroni, editors, *Declarative Agent Languages and Technologies III, Third International Workshop (DALT 2005)*, volume 3904 of *Lecture Notes in Computer Science*, pages 188–204. Springer, 2005.

[26] Mastercard and VISA. SET Secure Electronic Transaction Specification. 1997.

[27] R. Parikh and R. Ramanujam. Distributed processes and the logic of knowledge. In R. Parikh, editor, *Logics of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 256–268. Springer, 1985.

[28] R. Pucella. *Reasoning about Resource-Bounded Knowledge: Theory and Application to Security Protocol Analysis*. Ph.D. Thesis, Cornell University, 2004.

[29] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1), 1998.

[30] K. S. Ron van der Meyden. Symbolic model checking the knowledge of the dining cryptographers. In *Proc. CSFW'04*, 2004.

[31] A. Sabelfeld and A. C. Myers. Language-based information-flow security. *IEEE J. Selected Areas in Communications*, 21(1):5–19, Jan. 2003.

[32] P. F. Syverson and P. C. van Oorschot. A unified cryptographic protocol logic. NRL Publication 5540-227, Naval Research Lab, 1996.

[33] M.-J. Toussaint and P. Wolper. Reasoning about cryptographic protocols. In J. Feigenbaum and M. Merritt, editors, *Distributed Computing and Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 245–262. American Mathematical Society, 1989.